Mobile Applications

1. Short Description

The name of the application is YouPayTM.

The purpose of the application is to help the client keep track of all the fines and bills receive.

The application should allow the client to see all the fines and bills they have registered, along with filtering options for seeing which ones they have paid, and most importantly, which ones they still have to pay and when they are due.

1. Domain details.

Entity: Due Payment

Fields:

Total Sum: The total amount of money for the bill or fine.

Paid Sum: The sum of money paid so far.

Type: Bill or Fine

Reason: What the client has to pay for (ex: security system installation).

Due date: The date by which the bill or fine must be fully paid.

IBAN: The IBAN code through which the payment could be made. (where possible)

Address: The address to which the client could go to make their payment.

Institution: The name of the institution to which the sum of money must be paid (ex: ThiefProof SRL)

Status: Unpaid, Partially paid, Fully paid.

1. CRUD

Create:

1. To register a Due Payment, the client should click a ‘Plus’ button in the upper right corner of the screen.
2. They are taken to a separate screen and asked to select the Type of the Due Payment: either Bill or Fine.
3. A form is presented where the Total Sum, Due Date, and Reason must be specified, and are mandatory. Optionally, the fields Paid Sum, IBAN, Address, Institution may also be filled. The field Status is automatically set.
4. After the mandatory fields are completed, the ‘Save’ buttons is unlocked and the Due Payment can be saved.

Read:

1. When the client opens the app, he is presented a list with all the Due Payments which are not Fully Paid, in the order of their importance (past due payments, then payments farther and farther away from the due date).
2. From the top bar of the app, the client, by clicking on a label ‘Payments’, is brought to a screen where they can see all the Payments in the order in which they were registered (most recently registered first).
3. To go back to the previous screen (from point 1.), the client clicks on another label from the top bar ‘Due’.

Update:

1. To update the information of a Due Payment, the user holds the Due Payment which they wish to modify.

All the other fields, including Total Sum, can be modified.

1. Paid Sum must be between 0 and Total Sum. Status is automatically set. If Total Sum is modified to a lesser value and it is less that Paid Sum, Paid Sum is modified to have the same value as Total Sum, and the Status is set to ‘Fully Paid’.
2. The mandatory fields cannot be left empty, but the optional ones can be.

Delete:

1. To delete a Due Payment, click on the ‘X’ on the right side of the Due Payment. The list is immediately updated to reflect the deletion.
2. Persistence details

All the operations that modify the data are persisted on both the local db and the server: Create, Update, Delete.

1. Offline scenarios:
2. Create: Newly added Due Payments are stored on the local database. When the device reconnects to the server, the local database is synchronized with the server database.
3. Read: The application first attempts to fetch the data from the server. If it doesn’t receive a response after a few moments, then it attempts to read the data from the cache. It not all Due Payments are in the cache, then the user is asked to try again to connect to the server, and will not be shown any Due Payment.
4. Update: Updates are also made on the local database, and when connection to server is re-established, the two databases are synchronized.
5. Delete: Deleted Due Payments are also stored locally, with synchronization when server connection is re-established.